AMENDMENTS TO THE CLAIMS

Please amend claims 1, 11-21 and 31 as follows.

1. (Currently amended) A method comprising:

evaluating performance of a network having N optical links using a computer simulator, wherein evaluating the performance of the network includes generating random samples, comprising:

generating a first covariance matrix from a desired mean vector and a desired covariance matrix of a Bernoulli distribution, wherein the desired mean vector includes N elements and the desired covariance matrix has a dimension N x N where N is the number of optical links of the network;

constructing a normal vector using the desired mean vector and the first covariance matrix, wherein the normal vector includes N elements where N is the number of optical links of the network; and

generating a sampling vector using the normal vector and a threshold vector, wherein the sampling vector includes N elements where N is the number of optical links of the network, the sampling vector having the desired mean vector and the desired covariance matrix.

2. (Original) The method of claim 1 wherein generating the first covariance matrix comprises:

generating an integral expression F for a first non-diagonal element s_{ij} of the first covariance matrix at a row index i and a column index j, the integral expression having an integral limit as function of threshold elements τ_i and τ_j in the threshold vector at the vector indices i and j; and

obtaining the first non-diagonal element s_{ij} using the integral expression F, a mean μ_k of the desired mean vector, and a desired non-diagonal element Σ_{ij} of the desired covariance matrix.

3. (Original) The method of claim 2 further comprising:

obtaining a diagonal element s_{jj} of the first covariance matrix at a first row index j and a first column index j using the mean μ_j at the vector index j, the diagonal element being equal to a desired diagonal element Σ_{jj} of the desired covariance matrix.

4. (Original) The method of claim 3 further comprising:

generating a threshold element τ_j of the threshold vector at a vector index j, the threshold element being equal to $\mu_j + \sigma_j \sqrt{2}$ inverf$(1-2\mu_j)$ wherein μ_j and σ_j are desired mean and variance, respectively, at the vector index j and inverf is an inverse error function.

5. (Original) The method of claim 2 wherein constructing the normal vector comprises:

generating normal elements of the normal vector using the desired mean vector and the first covariance matrix.

6. (Original) The method of claim 5 wherein generating the sampling vector comprises:

comparing a normal element Y_k of the normal vector at a vector index k with a corresponding threshold element τ_k of the threshold vector at the vector index k;

setting a sampling element of the sampling vector at the vector index k to a first value if the normal element Y_k is greater than the corresponding threshold element τ_k; and

setting the sampling element of the sampling vector at the vector index k to a second value if the normal element Y_k is equal to or less than the corresponding threshold element τ_k.

7. (Original) The method of claim 2 wherein generating the integral expression F comprises:

forming a first variable $\rho = s_{ij}/(\sigma_i\sigma_j)$;

forming a second variable $c = \sqrt{2(1-\rho^2)}$;

forming a third variable $P = (a_i + a_j)/(c\sqrt{2})$, P being one of the integral limits;

forming a fourth variable $Q = (a_j - a_i)/(c\sqrt{2})$; and

forming the integral expression

$$F(\rho) = \frac{\sqrt{1-\rho}}{2\sqrt{\pi}} \int_P^{\infty} e^{-p^2(1-\rho)} (erf\sqrt{1+\rho}(Q+p+P) - erf(\sqrt{1+\rho}(Q-p+P)))dp$$

wherein p is an integral variable, erf is an error function, a_i and a_j are respectively equal to $(\tau_i-\mu_i)/\sigma_i$ and $(\tau_j-\mu_j)/\sigma_j$, τ_i and τ_j being the threshold elements at the vector indices equal respectively to the row index i and the column index j, μ_i and μ_j being the means at the vector indices equal respectively to the row index i and the column index j, σ_i and σ_j being the variances at the vector indices equal respectively to the row index i and the column index j.

8. (Original) The method of claim 7 wherein obtaining the first non-diagonal element comprises:

determining a right hand side (RHS) quantity $g_{ij} = \Sigma_{ij} + \mu_i\mu_j$;

equating the integral expression to the RHS quantity to form an integral equation $F = g_{ij}$; and

solving the integral equation for the first variable ρ.

9. (Original) The method of claim 8 wherein solving the integral equation comprises: solving the integral equation using a numerical method.

10. (Original) The method of claim 6 wherein the first value is 1 and the second value is 0.

11. (Currently amended) ~~A computer program product comprising:~~

~~a machine useable medium having program code embedded therein, the program code comprising:~~

An article of manufacture, comprising:

a machine-readable medium including a plurality of instructions which when executed perform operations comprising:

evaluating performance of a network having N optical links, wherein evaluating the performance of the network includes generating random samples, comprising:

~~computer readable program code to generate~~ generating a first covariance matrix from a desired mean vector and a desired covariance matrix of a Bernoulli distribution, wherein the desired mean vector includes N elements and the desired covariance matrix has a dimension N x N where N is the number of optical links of the network;

~~computer readable program code to construct~~ constructing a normal vector using the desired mean vector and the first covariance matrix, wherein the normal vector includes N elements where N is the number of optical links of the network; and

~~computer readable program code to generate~~ generating a sampling vector using the normal vector and a threshold vector, wherein the sampling vector includes N elements where N is the number of optical links of the network, the sampling vector having the desired mean vector and the desired covariance matrix.

12. (Currently amended) ~~The computer program product of claim 11 wherein the computer readable program code to generate the first covariance matrix comprises:~~

The article of manufacture of claim 11 wherein generating the first covariance matrix comprises:

~~computer readable program code to generate~~ generating an integral expression F for a first non-diagonal element s_{ij} of the first covariance matrix at a row index i and a column index j, the integral expression having an integral limit as function of threshold elements τ_i and τ_j in the threshold vector at the vector indices i and j; and

~~computer readable program code to obtain~~ obtaining the first non-diagonal element s_{ij} using the integral expression F, a mean μ_k of the desired mean vector, and a desired non-diagonal element Σ_{ij} of the desired covariance matrix.

13. (Currently amended) ~~The computer program product of claim 12 further comprising:~~

The article of manufacture of claim 12 further comprising:

~~computer readable program code to obtain~~ obtaining a diagonal element s_{jj} of the first covariance matrix at a first row index j and a first column index j using the mean μ_j at the vector index j, the diagonal element being equal to a desired diagonal element Σ_{jj} of the desired covariance matrix.

14. (Currently amended) ~~The computer program product of claim 13 further comprising:~~

The article of manufacture of claim 13 further comprising:

~~computer readable program code to generate~~ generating a threshold element τ_j of the threshold vector at a vector index j, the threshold element being equal to μ_j + $\sigma_j \sqrt{2}$ inverf(1-2μ_j) wherein μ_j and σ_j are desired mean and variance, respectively, at the vector index j and inverf is an inverse error function.

15. (Currently amended) ~~The computer program product of claim 12 wherein the computer readable program code to construct the normal vector comprises:~~

The article of manufacture of claim 12 wherein constructing the normal vector comprises:

~~computer readable program code to generate~~ generating normal elements of the normal vector using the desired mean vector and the first covariance matrix.

16. (Currently amended) ~~The computer program product of claim 15 wherein the computer readable program code to generate the sampling vector comprises:~~

The article of manufacture of claim 15 wherein generating the sampling vector comprises:

~~computer readable program code to compare~~ comparing a normal element Y_k of the normal vector at a vector index k with a corresponding threshold element τ_k of the threshold vector at the vector index k;

~~computer readable program code to set~~ setting a sampling element of the sampling vector at the vector index k to a first value if the normal element Y_k is greater than the corresponding threshold element τ_k; and

~~computer readable program code to set~~ setting the sampling element of the sampling vector at the vector index k to a second value if the normal element Y_k is equal to or less than the corresponding threshold element τ_k.

17. (Currently amended) ~~The computer program product of claim 12 wherein the computer readable program code to generate the integral expression F comprises:~~

The article of manufacture of claim 12 wherein generating the integral expression F comprises:

~~computer readable program code to form~~ forming a first variable $\rho = s_{ij}/(\sigma_i\sigma_j)$;

~~computer readable program code to form~~ forming a second variable $c = \sqrt{2(1-\rho^2)}$;

~~computer readable program code to form~~ forming a third variable $P = (a_i + a_j)/(c\sqrt{2})$, P being one of the integral limits;

~~computer readable program code to form~~ forming a fourth variable $Q = (a_j - a_i)/(c\sqrt{2})$; and

~~computer readable program code to form~~ forming the integral expression

$$F(\rho) = \frac{\sqrt{1-\rho}}{2\sqrt{\pi}} \int_{P}^{\infty} e^{-p^2(1-\rho)} (erf\sqrt{1+\rho}(Q+p+P) - erf(\sqrt{1+\rho}(Q-p+P)))dp$$

wherein p is an integral variable, erf is an error function, a_i and a_j are respectively equal to $(\tau_i-\mu_i)/\sigma_i$ and $(\tau_j-\mu_j)/\sigma_j$, τ_i and τ_j being the threshold elements at the vector indices equal respectively to the row index i and the column index j, μ_i and μ_j being the means at the vector indices equal respectively to the row index i and the column index j, σ_i and σ_j being the variances at the vector indices equal respectively to the row index i and the column index j.

18. (Currently amended) ~~The computer program product of claim 17 wherein the computer readable program code to obtain the first non-diagonal element comprises:~~

The article of manufacture of claim 17 wherein obtaining the first non-diagonal element comprises:

~~computer readable program code to determine~~ determining a right hand side (RHS) quantity $g_{ij} = \Sigma_{ij} + \mu_i\mu_j$;

~~computer readable program code to equate~~ equating the integral expression to the RHS quantity to form an integral equation $F = g_{ij}$; and

~~computer readable program code to solve~~ solving the integral equation for the first variable ρ.

19. (Currently amended) ~~The computer program product of claim 18 wherein the computer readable program code to solve the integral equation comprises~~:

The article of manufacture of claim 18 wherein solving the integral equation comprises:

~~computer readable program code to solve~~ solving the integral equation using a numerical method.

20. (Currently amended) ~~The method of claim 16~~ The article of manufacture of claim 16 wherein the first value is 1 and the second value is 0.

21. (Currently amended) A network simulator for evaluating performance of a network of free-space optical links, wherein the network simulator comprises:

a network modeler to model [[a]] the network of free-space optical links;

a reliability modeler coupled to the network modeler to evaluate a reliability model for the network; and

a random sampler coupled to the network modeler and the reliability modeler to generate random samples for a Bernoulli distribution, the random sampler comprising:

a covariance generator to generate a first covariance matrix from a desired mean vector and a desired covariance matrix of the Bernoulli distribution,

a normal vector generator coupled to the covariance generator to construct a normal vector using the desired mean vector and the first covariance matrix, and

a thresholder coupled to the covariance generator and the normal vector generator to generate a sampling vector using the normal vector and a threshold vector, the sampling vector having the desired mean vector and the desired covariance matrix.

22. (Original) The simulator of claim 21 wherein the covariance generator comprises:

an integral expression generator to generate an integral expression F for a first non-diagonal element s_{ij} of the first covariance matrix at a row index i and a column index j, the integral expression having an integral limit as function of threshold elements τ_i and τ_j in the threshold vector at the vector indices i and j; and

a non-diagonal element generator coupled to the integral expression generator to obtain the first non-diagonal element s_{ij} using the integral expression F, a mean μ_k of the desired mean vector, and a desired non-diagonal element Σ_{ij} of the desired covariance matrix.

23. (Original) The simulator of claim 22 wherein the random sampler further comprises:

a diagonal element generator to obtain a diagonal element s_{jj} of the first covariance matrix at a first row index j and a first column index j using the mean μ_j at the vector index j, the diagonal element being equal to a desired diagonal element Σ_{jj} of the desired covariance matrix.

24. (Original) The simulator of claim 23 wherein the random sampler further comprises:

a threshold vector calculator coupled to the first normal vector generator to generate a threshold element τ_j of the threshold vector at a vector index j, the threshold element being equal to $\mu_j + \sigma_j\sqrt{2}$ inverf$(1\text{-}2\mu_j)$ wherein μ_j and σ_j are the desired mean and variance, respectively, at the vector index j and inverf is an inverse error function.

25. (Original) The simulator of claim 22 wherein the normal vector generator generates normal elements of the normal vector using the desired mean vector and the first covariance matrix.

26. (Original) The simulator of claim 25 wherein thresholder comprises:

a comparator to compare a normal element Y_k of the normal vector at a vector index k with a corresponding threshold element τ_k of the threshold vector at the vector index k; and

a selector coupled to the comparator to set a sampling element of the sampling vector at the vector index k to a first value if the normal element Y_k is greater than the corresponding threshold element τ_k and to set the sampling element of the sampling vector at the vector index k to a second value if the normal element Y_k is equal to or less than the corresponding threshold element τ_k.

27. (Original) The simulator of claim 22 wherein the integral expression generator generates the integral expression

$$F(\rho) = \frac{\sqrt{1-\rho}}{2\sqrt{\pi}} \int_P^{\infty} e^{-p^2(1-\rho)} (erf\sqrt{1+\rho}(Q+p+P) - erf(\sqrt{1+\rho}(Q-p+P)))dp$$

wherein:

$\rho = s_{ij}/(\sigma_i\sigma_j)$, $c = \sqrt{2(1-\rho^2)}$, $P = (a_i + a_j)/(c\sqrt{2})$, P being one of the integral limits, $Q = (a_j - a_i)/(c\sqrt{2})$, p is an integral variable, erf is an error function, a_i and a_j are respectively equal to $(\tau_i-\mu_i)/\sigma_i$ and $(\tau_j-\mu_j)/\sigma_j$, τ_i and τ_j being the threshold elements at the vector indices equal respectively to the row index i and the column index j, μ_i and μ_j being the means at the vector indices equal respectively to the row index i and the column index j, σ_i and σ_j being the variances at the vector indices equal respectively to the row index i and the column index j.

28. (Original) The simulator of claim 27 wherein the non-diagonal element generator comprises:

a right hand side (RHS) generator to determines a right hand side (RHS) quantity $g_{ij} = \Sigma_{ij} + \mu_I\mu_j$;

an equation solver coupled to the integral expression generator and the RHS generator to equate the integral expression to the RHS quantity to form an integral equation $F = g_{ij}$, and to solve the integral equation for the first variable ρ.

29. (Original) The simulator of claim 28 wherein the equation solver solves the integral equation using a numerical method.

30. (Original) The simulator of claim 26 wherein the first value is 1 and the second value is 0.

31. (Currently amended) A system comprises:

a processor; and

a memory coupled to the processor, the memory having program code to evaluate performance of a network having N optical links, the program code when executed by the processor causing the processor to:

generate random samples, comprising:

generate a first covariance matrix from a desired mean vector and a desired covariance matrix of a Bernoulli distribution, wherein the desired mean vector includes N elements and the desired covariance matrix has a dimension N x N where N is the number of optical links of the network,

construct a normal vector using the desired mean vector and the first covariance matrix, wherein the normal vector includes N elements where N is the number of optical links of the network, and

generate a sampling vector using the normal vector and a threshold vector, wherein the sampling vector includes N elements where N is the number of optical links of the network, the sampling vector having the desired mean vector and the desired covariance matrix.

32. (Original) The system of claim 31 wherein the program code causing the processor to generate the first covariance matrix causes the processor to:

generate an integral expression F for a first non-diagonal element s_{ij} of the first covariance matrix at a row index i and a column index j, the integral expression having an integral limit as function of threshold elements τ_i and τ_j in the threshold vector at the vector indices i and j; and

obtain the first non-diagonal element s_{ij} using the integral expression F, a mean μ_k of the desired mean vector, and a desired non-diagonal element Σ_{ij} of the desired covariance matrix.

33. (Original) The system of claim 32 wherein the program code, when executed, further causing the processor to:

obtain a diagonal element s_{jj} of the first covariance matrix at a first row index j and a first column index j using the mean μ_j at the vector index j, the diagonal element being equal to a desired diagonal element Σ_{jj} of the desired covariance matrix.

34. (Original) The system of claim 33 wherein the program code, when executed, further causing the processor to:

generate a threshold element τ_j of the threshold vector at a vector index j, the threshold element being equal to $\mu_j + \sigma_j\sqrt{2}$ inverf(1-2μ_j) wherein μ_j and σ_j are desired mean and variance, respectively, at the vector index j and inverf is an inverse error function.

35. (Original) The system of claim 32 wherein the program code causing the processor to construct the normal vector causes the processor to:

generate normal elements of the normal vector using the desired mean vector and the first covariance matrix.

36. (Original) The system of claim 35 wherein the program code causing the processor to generate the sampling vector causes the processor to:

compare a normal element Y_k of the normal vector at a vector index k with a corresponding threshold element τ_k of the threshold vector at the vector index k;

set a sampling element of the sampling vector at the vector index k to a first value if the normal element Y_k is greater than the corresponding threshold element τ_k; and

set the sampling element of the sampling vector at the vector index k to a second value if the normal element Y_k is equal to or less than the corresponding threshold element τ_k.

37. (Original) The system of claim 32 wherein the program code causing the processor to generate the integral expression F causes the processor to:

form a first variable $\rho = s_{ij}/(\sigma_i\sigma_j)$;

form a second variable $c = \sqrt{2(1-\rho^2)}$;

form a third variable $P = (a_i + a_j)/(c\sqrt{2})$, P being one of the integral limits;

form a fourth variable $Q = (a_j - a_i)/(c\sqrt{2})$; and

form the integral expression

$$F(\rho) = \frac{\sqrt{1-\rho}}{2\sqrt{\pi}} \int_{P}^{\infty} e^{-p^2(1-\rho)} (erf\sqrt{1+\rho}(Q+p+P) - erf(\sqrt{1+\rho}(Q-p+P)))dp$$

wherein p is an integral variable, erf is an error function, a_i and a_j are respectively equal to $(\tau_i-\mu_i)/\sigma_i$ and $(\tau_j-\mu_j)/\sigma_j$, τ_i and τ_j being the threshold elements at the vector indices equal respectively to the row index i and the column index j, μ_i and μ_j being the means at the vector indices equal respectively to the row index i and the column index j, σ_i and σ_j being the variances at the vector indices equal respectively to the row index i and the column index j.

38. (Original) The system of claim 32 wherein the program code causing the processor to obtain the first non-diagonal element causes the processor to:

determine a right hand side (RHS) quantity $g_{ij} = \Sigma_{ij} + \mu_i\mu_j$;

equate the integral expression to the RHS quantity to form an integral equation $F = g_{ij}$; and

solve the integral equation for the first variable ρ.

39. (Original) The system of claim 38 wherein the program code causing the processor to solve the integral equation causes the processor to:

solve the integral equation using a numerical method.

40. (Original) The system of claim 36 wherein the first value is 1 and the second value is 0.